

DETECTION OF ARP SPOOFING

A THESIS

Submitted by

Shahasiya Muhammed AC (RCAS2021MCS208)

in partial fulfillment for the award of the degree of

**MASTER OF SCIENCE
SPECIALIZATION IN
INFORMATION SECURITY AND CYBER FORENSICS**



**DEPARTMENT OF COMPUTER SCIENCE
RATHINAM COLLEGE OF ARTS AND SCIENCE
(AUTONOMOUS)**

COIMBATORE - 641021 (INDIA)

MAY-2023

RATHINAM COLLEGE OF ARTS AND SCIENCE
(AUTONOMOUS)
COIMBATORE - 641021



BONAFIDE CERTIFICATE

This is to certify that the thesis entitled **Detection of ARP Spoofing** submitted by **Shahasiya Muhammed AC,,** for the award of the Degree of Master in Computer Science specialization in **“INFORMATION SECURITY AND CYBER FORENSICS”** is a bonafide record of the work carried out by her under my guidance and supervision at Rathinam College of Arts and Science, Coimbatore

Mr. Saravana Kumar
Supervisor

Dr.P.Sivaprakash
Mentor

Submitted for the University Examination held on 09.05.2023

INTERNAL EXAMINER

EXTERNAL EXAMINER

RATHINAM COLLEGE OF ARTS AND SCIENCE
(AUTONOMOUS)
COIMBATORE - 641021

DECLARATION

I, **Shahasiya Muhammed AC**, hereby declare that this Phase 1 entitled "**Detection of ARP Spoofing**", is the record of the original work done by us under the guidance of **Mr. Saravana Kumar**, Faculty Rathinam college of arts and science, Coimbatore. To the best of my knowledge this work has not formed the basis for the award of any degree or a similar award to any candidate in any University.

Signature of the Students:

Shahasiya Muhammed AC

Place: Coimbatore

Date: 09.05.2023

COUNTERSIGNED

Mr. Saravana Kumar
Supervisor

Contents

Acknowledgement	iv
List of Figures	v
List of Abbreviations	vi
Abstract	vii
1 Introduction	1
1.1 Objective of the project	3
1.2 Scope of the Project	4
1.3 Module Description	5
1.4 Existing System	6
2 Literature Survey	8
2.1 SDN paradigm:	8
2.2 Modified ICMP and Voting:	9
2.3 Gratuitous Decision Packet	10
2.4 RTNSS	11

3	Methodology	12
3.1	Implementaion and working process:	12
3.2	Software and Hardware Requirements:	14
3.3	Technologies Used:	14
3.4	System Design	15
4	Experimental Setup	17
4.1	Create Intent	17
4.2	Proposed Algorithm	18
4.2.1	Get user Input	18
4.2.2	Get User Response	20
4.3	User Interface	20
5	Result and Discussions	22
6	Deployment Process	23
6.1	Overview of the process	23
6.2	Algorithm Implementation	25
6.3	Component testing	25
6.3.1	Network Traffic Interception	25
6.3.2	ARP traffic Filter	26
6.3.3	ARP Spoof Detetctor:	28
6.3.4	Alert Administrator	28

7 Conclusion	32
7.1 Future Works	33
References	34

Acknowledgement

On successful completion for project look back to thank who made in possible. First and foremost, thank “**THE ALMIGHTY**” for this blessing on me without which I could have not successfully my project. I am extremely grateful to **Dr.Madan.A. Sendhil, M.S., Ph.D.**, Chairman, Rathinam Group of Institutions, Coimbatore and **Dr. R.Manickam MCA., M.Phil., Ph.D.**, Secretary, Rathinam Group of Institutions, Coimbatore for giving me opportunity to study in this college. I am extremely grateful to **Dr.R.Muralidharan, M.Sc., M.Phil., M.C.A., Ph.D.**, Principal Rathinam College of Arts and Science(Autonomous), Coimbatore. Extend deep sense of valuation to **Mr.A.Uthiramoorthy, M.C.A., M.Phil., (Ph.D)**, Rathinam College of Arts and Science (Autonomous) who has permitted to undergo the project.

Unequally I thank **Dr.P.Sivaprakash, M.E., Ph.D.**, Mentor and **Dr.Mohamed Mallick, M.E., Ph.D.**, Project Coordinator, and all the Faculty members of the Department - iNurture Education Solution pvt ltd for their constructive suggestions, advice during the course of study. I convey special thanks, to the supervisor **Mr.Saravana Kumar.**, who offered their inestimable support, guidance, valuable suggestion, motivations, helps given for the completion of the project.

I dedicated sincere respect to my parents for their moral motivation in completing the project.

List of Figures

3.1	Overview of Proposed system	16
4.1	Experimental Setup	17
4.2	Attacker Machine	19
4.3	Victim Machine	20
4.4	User interface	21
6.1	Network Traffic logs	26
6.2	Filtered ARP Traffic logs	27
6.3	Files of ARP Spoof detector	29
6.4	ARP Spoof detector in Victims machine	29
6.5	network Traffic Interceptor	30
6.6	ARP Traffic Interceptor	30
6.7	Attackers machine	31
6.8	Alert on terminal	31
6.9	Alert via Email	31

List of Abbreviations

ARP	Address Resolution Protocol
LAN	Local Area Network
IP	Internet protocol
MAC	Media Access Control
RAM	Random Access Memory
NAT	Network Address Translation
TCP	Transmission control Protocol
SDN	Software Design Networks

Abstract

Address Resolution Protocol (ARP) spoofing is a type of Cyber-attack carried out over a local area network (LAN) that involves sending malicious ARP packets to a default gateway on a LAN in order to change the pairings in its Internet Protocol (IP) to Media Access Control (MAC) address table. ARP protocol translates IP addresses into MAC addresses because the ARP protocol was designed purely for efficiency and not for security. As long as the attacker is directly connected to the target LAN or in control of a workstation within it, ARP spoofing attacks are very simple to execute. The attack is carried out by an attacker deceiving the default network gateway into believing that their MAC address should be linked to the target's IP address (and vice versa, making the target's MAC address linked to the attacker's IP address). All of the target's traffic to any other device on the network travels through the attacker's computer after the default gateway receives this message and broadcasts its modifications to all other devices on the network, enabling the attacker to inspect or alter it before forwarding it to its actual destination. Since ARP spoofing attacks occur on such a low level, users targeted by ARP spoofing rarely realize that their traffic is being inspected or modified besides man-in-the-middle attack. ARP spoofing can be used to cause a

denial-of-service condition over a LAN by simply intercepting or dropping and not forwarding the target's packets. The ARP spoofing attack can be prevented by this email alert notification which is sent to the administrator, so that the user can be able to know that the attack is happening and the communication can be terminated and the user can stop the attack from happening.

Chapter 1

Introduction

Address Resolution Protocol (ARP) spoofing is a type of cyber-attack carried out over a local area network (LAN) that involves sending malicious ARP packets to a default gateway on a LAN in order to change the pairings in its Internet Protocol (IP) to Media Access Control (MAC) address table. ARP protocol translates IP addresses into MAC addresses because the ARP protocol was designed purely for efficiency and not for security. As long as the attacker is directly connected to the target LAN or in control of a workstation within it, ARP spoofing attacks are very simple to execute. The attack is carried out by an attacker deceiving the default network gateway into believing that their MAC address should be linked to the target's IP address (and vice versa, making the target's MAC address linked to the attacker's IP address). Once the default gateway has received this message and broadcasts its changes to all other devices on the network, all of the target's traffic to any other device on the network travels through the attacker's computer, allowing the attacker to inspect or modify it before forwarding it to its real destination. Since ARP spoofing attacks occur on such a low level, users targeted by ARP spoofing rarely realize that their traffic is being

inspected or modified besides man-in-the-middle attack. ARP spoofing can be used to cause a denial-of-service condition over a LAN by simply intercepting or dropping and not forwarding the target's packets. Most of the organizations implement LAN for their communication and networking needs. The MAC address is the communication identifier in LANs. So, in order to transport packets and communicate within a LAN, IP addresses must be resolved to MAC addresses. The Address Resolution Protocol handles this resolution (ARP). This protocol, however, has a serious security flaw. The ARP protocol has no state. It doesn't verify whether a request was ever made for the response that was given. As a result, it is vulnerable to an exploit called an ARP Spoofing attack. As long as the attacker is directly connected to the target LAN or in control of a workstation within it, ARP spoofing attacks are very simple to execute. The attack is carried out by an attacker deceiving the default network gateway into believing that their MAC address should be linked to the target's IP address (and vice versa, making the target's MAC address linked to the attacker's IP address). The ARP spoofing attack can be prevented by this email alert notification which is sent to the administrator, so that the user can be able to know that the attack is happening and the communication can be terminated and the user can stop the attack from happening.

1.1 Objective of the project

The main objective the project is to address two issues; first, is to identify the attacker and second, is to prevent the ARP Spoofing attack. The attacker will spoof the MAC address to perform ARP spoofing. The goal here would be to check for spoofed MAC address and to prevent the attack by notifying the administrator via terminal and the email

For their networking and communication requirements, the majority of enterprises use LAN. The MAC address is the communication identifier in LANs. So, in order to transport packets and communicate within a LAN, IP addresses must be resolved to MAC addresses. The Address Resolution Protocol handles this resolution (ARP). This protocol, however, has a serious security flaw. The ARP protocol has no state. It doesn't verify whether a request was ever made for the response that was given. As a result, it is more vulnerable to an exploit called an ARP spoofing attack.

LAN communication between two hosts can now be tapped into very easily thanks to software that can be downloaded from the Internet. These instruments make advantage of the Address Resolution Protocol (ARP) spoofing technique, which depends on hosts storing reply messages even though the relevant requests weren't ever received.

The attacker launches the attack by sending a bogus ARP reply message to the default network gateway, notifying it that his or her MAC address should be associated with the target's IP address (and vice versa, so the target's MAC address is now associated with the attacker's IP address). Once the default gateway receives this message and

broadcasts its changes to all other network devices, all traffic from the target to any other network device passes through the attacker's computer, allowing the attacker to inspect or modify it before forwarding it to its true destination. Because ARP spoofing attacks occur at such a low level, consumers who are targeted by ARP spoofing are rarely aware that their traffic is being reviewed or manipulated.

1.2 Scope of the Project

This project is being developed with the aim to demonstrate and detect ARP Spoofing. To perform Man-in-the-middle (MITM) attacks, ARP spoofing is mostly used. Being able to witness few ARP Spoofing cases, was the principal reason of inspiration to find a solution. The plan is to prepare a script to detect an ARP Spoofing. The outcome of this project. The goal of this project is to identify and show ARP spoofing. ARP spoofing is mainly employed in Man-in-the-middle (MITM) attacks. The main source of inspiration for developing a solution was being able to see a few instances of ARP spoofing. A script will be written in order to recognise an ARP spoofing. The project's result would be that the administrator might be informed in order to stop the attack.

Already different methods have been proposed to transfer data securely over the internet. However, none of them has provided a feasible solution to counter the ARP spoofing. Thanks to tools that can be downloaded from the Internet, intercepting the communication between two hosts on a LAN has become quite simple. These instruments employ the Address Resolution Protocol (ARP) spoofing method, which depends on hosts caching reply messages even though the corresponding requests weren't ever

sent.

False positives, false negatives, and continual maintenance are potential drawbacks of this strategy. Since ARP spoofing can take many different forms, detection could result in a high number of false alarms, which might cause incident response teams to ignore or suppress alarms without conducting a thorough investigation.

The goal of the proposed system is to illustrate and identify ARP spoofing while also alerting the administrator in the event that an attack occurs in real-time. For this project, there are many modules or features, including a module to notify the administrator, ARP Sniffer, ARP Spoof Detector, and Network ARP Traffic.

1.3 Module Description

Network Traffic Interceptor: Network traffic is the amount of data, which moves across a network during any given time. Network traffic is the main component for the network traffic measurement, network traffic control and simulation.

ARP Sniffer: A sniffer is a software or hardware tool that allows the user to “sniff” or monitor the internet or network traffic real time, capturing all the data flowing to and from the user’s computer.

ARP Spoof Detector: All devices are uniquely identified by a Media Access Control (MAC) address. Legitimate MAC addresses can easily be spoofed and used for various attacks such as MITM and DOS attacks. In ARP Spoofing attacks, MAC Address plays an important role.

Alert Administrator: Email alert/notification is an email sent to inform the administrator about any spoofed or updates in the ARP spoofing attacks

1.4 Existing System

SDN paradigm: Software Defined Network Authors: : Mohammad Z. Masoud, Yousf Jaradat and Ismael Jannoud Year: 2015. A new algorithm has been proposed and implemented to prevent this attack and an experiment has been conducted to evaluate the proposed algorithm. The result demonstrated that the new algorithm prevented ARP spoofing and many breaches and attacks utilizing ARP spoofing The centrality found in SDN so that any application will be deployed in the controller and that will affect all users of the network. The centrality found in SDN so that any application will be deployed in the controller and that will affect all users of the network.

Modified ICMP and Voting: Authors: Prerna Arote; Karam Veer Arya. Year: 2015. Description: The basic idea of presented architecture is to design such structure on more than three systems, for transmitting ARP and ICMP packets, or the central server plays an important role in proposed scheme. During the failure of any one of the systems, other system can work alone. Therefore, backward compatibility is achieved with the original ARP structure.

ASD: ARP Spoofing Detector using OpenWrt: Year:2022 . Description:Three information tables, AssocList, ARP cache table, and DHCP table, which are frequently managed by the access point based on a Linux system, are used by the access point-based ARP Spoofing Detector (ASD) to distinguish between ARP spoofing and con-

nections from VM guests.

ARP Poisoning Detection and Prevention using Scapy: Year:2021. Description: A Python program employing the Scapy module was used to create the ARP Spoof attack tool. For detecting the above-generated ARP Poisoning attack (or any ARP Poisoning attack in general), a detection algorithm has been proposed, and the algorithm has been implemented using a python script with the scapy library.

Chapter 2

Literature Survey

The “ARP spoofing” attack is based on impersonating a system in the network, making the two ends of a communication believe that the other end is the attacker’s system, intercepting the traffic interchanged. Numerous strategies have already been put out for safely transferring data over the internet. But none of them has offered a workable defence against ARP spoofing. Thanks to tools that can be obtained from the Internet, intercepting the traffic between two hosts on a LAN has become relatively simple. These instruments make use of the Address Resolution Protocol (ARP) spoofing technique, which depends on hosts caching response messages even though the corresponding requests weren’t sent.

2.1 SDN paradigm:

SDN has been utilized to alleviate and eliminate the problem of ARP poisoning attack. Many other network attacks, including man-in-the-middle, denial-of-service, and session hijacking, have this attack as their foundation. Software Defined Network (SDN) is an emerging network paradigm that introduces a centralized controller to the network.

This controller has the ability to manipulate and program the algorithms and protocols that run over the underlying hardware.

The ability of the SDN approach is to prevent ARP spoofing attack without modifying the original ARP protocol. The centrality found in SDN so that any application will be deployed in the controller and that will affect all users of the network. If the network's users obtain their IP addresses from a DHCP server, SDN controller can inspect DHCP reply messages to obtain the gateway IP address sent to the users in the option field of the message. The main drawbacks of this SDN paradigm was they were focused only on IPv4 and To the SDN structure, No new field will be added to the ARP messages.

2.2 Modified ICMP and Voting:

The basic idea of presented architecture is to design such structure on more than three systems, for transmitting ARP and ICMP packets, or the central server plays an important role in proposed scheme. During the failure of any one of the systems, other system can work alone. Therefore, backward compatibility is achieved with the original ARP structure

The main features of modified ICMP and Voting are it is less costly and easy to use but the main drawback of this system was To maintain fairness among different nodes where computational power of machines is more diverse. The algorithm and the technology used were router/switch and ARP packets

Possible weaknesses in this approach include false positives, false negatives, and ongoing maintenance. Since ARP spoofing can present itself in many forms, detection

could cause excessive false alarms, which could lead to incident response teams ignoring or suppressing alarms without proper investigation. From the literature review, analysis were done and the study revealed various methods used to detect and prevent ARP Spoofing attacks. There are many advantages as well as disadvantages to the above mentioned researches. To mitigate the drawbacks present in existing systems the solution proposed is to create a script, which could effectively detect the ARP Spoofing in real time scenario. This will help in mitigation of false alerts/alarms, and provide better results to the organization and in turn help maintain their security posture with respect to the ARP Spoofing attacks

2.3 Gratuitous Decision Packet

The two major objectives of the Gratuitous Decision Packet System are the detection of suspicious ARP packets via the use of real-time analysis of received ARP packets and the differentiation between a trustworthy and malicious host through the use of modified gratuitous ARP packet request packets.

The administrator must process the ARP cache using specialised network tools in order for the ARP watch and ARP Guard to function because they are manual solutions. This technique entails setting up VLAN (Virtual LAN), assigning static IP addresses to each host in the LAN, and other steps. This method is difficult for administrators to use, there is no way to distinguish between a malicious and legitimate host, and it is not appropriate for DHCP setups.

2.4 RTNSS

A server and an agent make up RTNSS. On a user's computer, an agent is installed, helping to defend it against ARP spoofing attacks. The server manages the information of the Agents, preventing an ARP spoofing attack and helping the Agent get a protection order by evaluating the ARP spoofing attack based on the information the Agent provided.

The system does not perform structural changes in the network or place a stress on the system and the network, and it delivers security and efficiency by overcoming the disadvantages of existing proposed models, such as encryption complexity.

Chapter 3

Methodology

3.1 Implementaion and working process:

This project can be implemented with a real network environment consisting of three systems (attacker, victim and gateway). VMware workstation pro is setup using NAT gateway where both the systems have Kali Linux distribution installed. All the systems are connected over wired network. The attacker's system is configured to have 1 GB RAM, 60 GB Harddisk and a processor whereas the victim's system have 2 GB RAM, 60 GB Harddisk and a processor.

The UI used for this project is the command-line interface. A command-line interface (CLI) is a text-based user interface (UI) used to run programs, manage computer files and interact with the computer. CLIs accept as input commands that are entered by keyboard; the computer then runs the commands invoked at the command prompt. The project is designed in such a way that the UI and the other elements will be user friendly. The programming languages used for this project is Python. A command line shell, Bash is used to run the scripts. The benefits of using shell is that it enables users

to write programs that can be run on the system from the command line, direct the output of one program to be the input for another program, and system variables which can be set at the command line, or the values of those variables displayed.

Algorithm Implementation

The current algorithm deals with 3 systems Victim, Attacker, and Gateway over the network connected through the wired connection. Attacker uses arpspoof tools to play the role of man in the middle like an intruder. Algorithm checks the ARP cache. With the help of Arpspoof tool, ARP poisoning is done between Victim and Gateway using arpspoof command. Now all the traffic is routed to an Attacker. It captures all kind of traffic including ARP request and ARP response packets flowing among hosts. The algorithm detects the spoof and alerts the administrator. Additionally, the network and ARP traffic is monitored and logged to a log file which will be saved in the server (here, victim).

- * Programming Language: Python

- * Scripting Language: Bash

- * Tools Used: Tshark, Tcpdump, Wireshark, Arpspoof

3.2 Software and Hardware Requirements:

Hardware Software Requirements:

- * Laptop with 4 GB RAM
- * Preferred Linux distro: Kali Linux 2021.4
- * Python Version: 3.9.0
- * A Router

3.3 Technologies Used:

Scapy:

Scapy is a library supported by both Python2 and Python3. It is used for interacting with the packets on the network. It includes a number of features that make it simple to counterfeit and tamper with the packet. Scapy's module allows us to build a variety of network tools, including packet dumpers, network scanners, and ARP spoofers. Using this module, more complex tools for network security and ethical hacking can be produced.

Netifaces:

It is a portable third-party library in Python to enumerate network interfaces on local machine. Historically it has been difficult to straightforwardly get the network addresses of the machine on which the Python scripts are running without compromising the portability of your script. Netifaces takes care of enumerating interfaces, network addresses and preserves the portability.

TCP dump:

Tcp dump is a data-network packet analyzer computer program that runs under a command line interface. It enables the user to view TCP/IP and other packets that are sent across a network to which the computer is connected, as well as other packets. Tcpdump is free software that is distributed under the BSD licence. Most Unix-like operating systems support Tcpdump. Tcpdump utilises the libpcap library in those systems to collect packets. WinDump, a port of tcpdump for Windows, makes use of WinPcap, the Windows implementation of libpcap.

Tshark:

TShark is a network protocol analyzer. It is the terminal version of Wireshark that supports similar options and is more useful. It lets the user capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file. TShark's native capture file format is pcapng format, which is also the format used by Wireshark and various other tools. This CLI is more easy to use than GUI.

3.4 System Design

Network traffic is the amount of data, which moves across a network during any given time. Network traffic is the main component for the network traffic measurement, network traffic control and simulation. A sniffer is a software or hardware tool that allows the user to “sniff” or monitor the internet or network traffic real time, capturing all the data flowing to and from the user's computer. All devices are uniquely identified

by a Media Access Control (MAC) address. Legitimate MAC addresses can easily be spoofed and used for various attacks such as MITM and DOS attacks. In ARP Spoofing attacks, MAC Address plays an important role. Email alert/notification is an email sent to inform the administrator about any spoofed or updates in the ARP spoofing attacks.

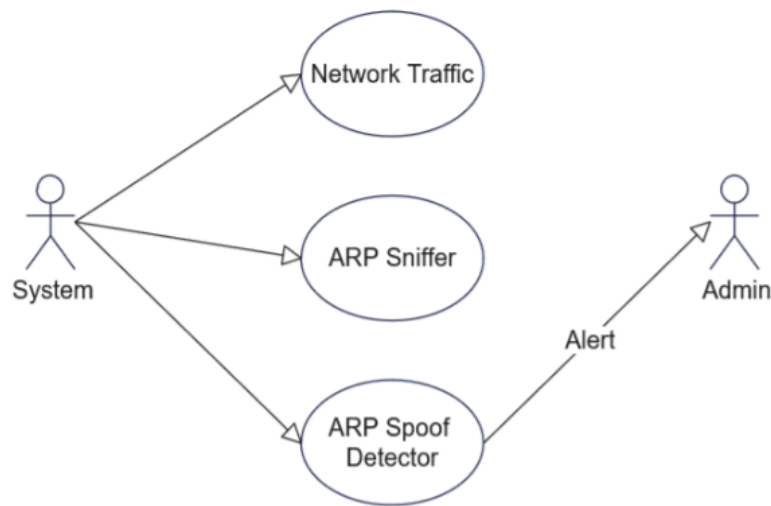


Figure 3.1: Overview of Proposed system

Chapter 4

Experimental Setup

4.1 Create Intent

Three systems (attacker, victim, and gateway) make up an actual network environment in which this project can be implemented. Both workstations have the Kali Linux distribution installed, and VMware Workstation Pro is configured utilising a NAT gateway. All the systems are connected over wired network. The victim's machine has 2 GB RAM, 60 GB Hard disk, and a processor, whereas the attacker's system only has 1 GB RAM and 60 GB Hard disk.

System	IP Address	MAC Address
Gateway	192.168.111.2	00:50:56:f6:8a:5e
Attacker	192.168.111.128	00:0c:29:c1:6d:91
Victim	192.168.111.129	00:0c:29:e3:43:0b

Figure 4.1: Experimental Setup

4.2 Proposed Algorithm

1. Start
2. Fire up the script
3. Import necessary packages and modules
4. Intercept Network Traffic
5. Filter ARP Traffic
6. Select an interface
7. Monitor the selected interface
8. If ARP Spoof attack detected, print "Attack Detected"
9. Alert the administration via email.
10. Go back to 6
11. If "Ctrl + C" is pressed, end the script.
12. End

4.2.1 Get user Input

In this approach, two Linux machines have created separately and those machines need to be assigned in same network which is NAT network. After that two machines need to be pinged using the ping command. After pinging the two machines each other, now the MAC address will be stored in ARP table. This ARP table will show the all previously connected machines. So by observing the MAC address in the ARP table, we can come into point that two machines are connected each other

Commands:

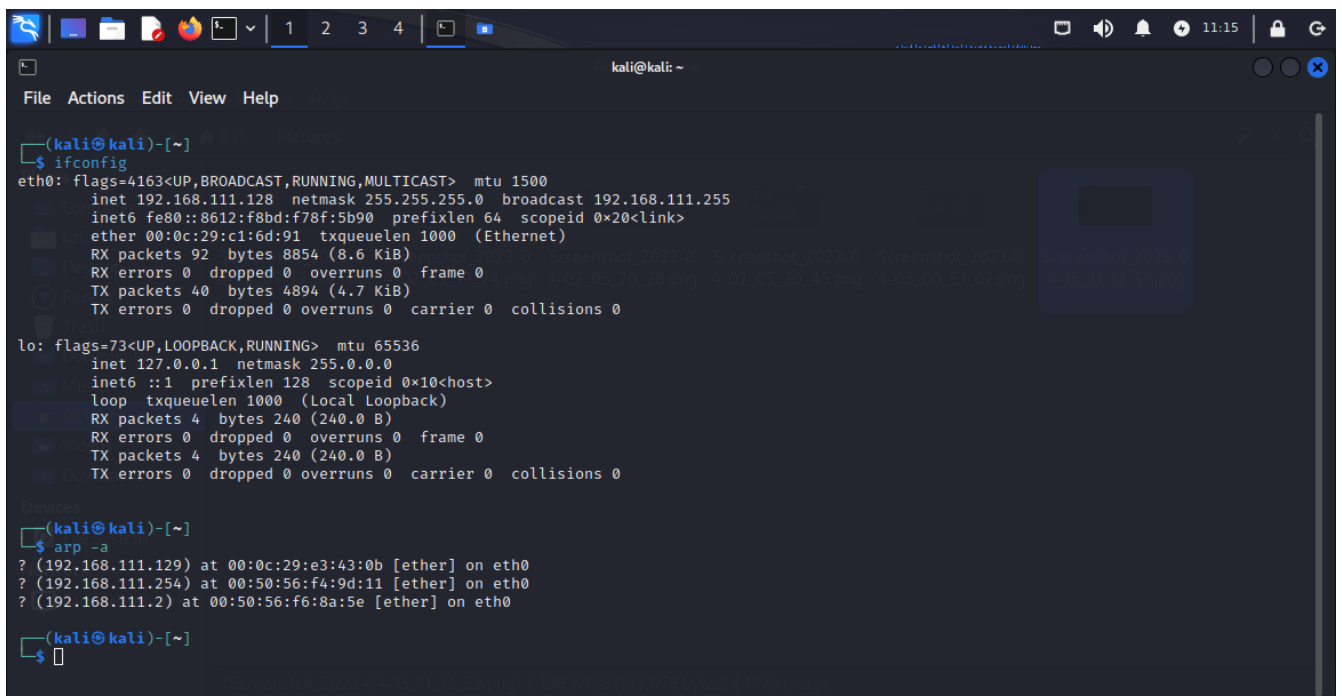
* **ifconfig** - To check the IP address and the MAC address of the particular system and the interfaces which the machine is connected.

* **ping**- Command to start communication between the two machines.

* **arp -a**- Command to check the ARP table to display the MAC address of the machines which connected.

***arpsoof**- tool used for ARP spoofing attack

***arp spoof -i eth0 -t 192.168.111.129 -r 192.168.111.2** - -i = interface, -t = target, -r = router/gateway



```
(kali@kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.111.128 netmask 255.255.255.0 broadcast 192.168.111.255
    inet6 fe80::8612:f8bd:f78f:5b90 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:c1:6d:91 txqueuelen 1000 (Ethernet)
    RX packets 92 bytes 8854 (8.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 40 bytes 4894 (4.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Docker:
(kali@kali)-[~]
└─$ arp -a
? (192.168.111.129) at 00:0c:29:e3:43:0b [ether] on eth0
? (192.168.111.254) at 00:50:56:f4:9d:11 [ether] on eth0
? (192.168.111.2) at 00:50:56:f6:8a:5e [ether] on eth0

(kali@kali)-[~]
└─$
```

Figure 4.2: Attacker Machine

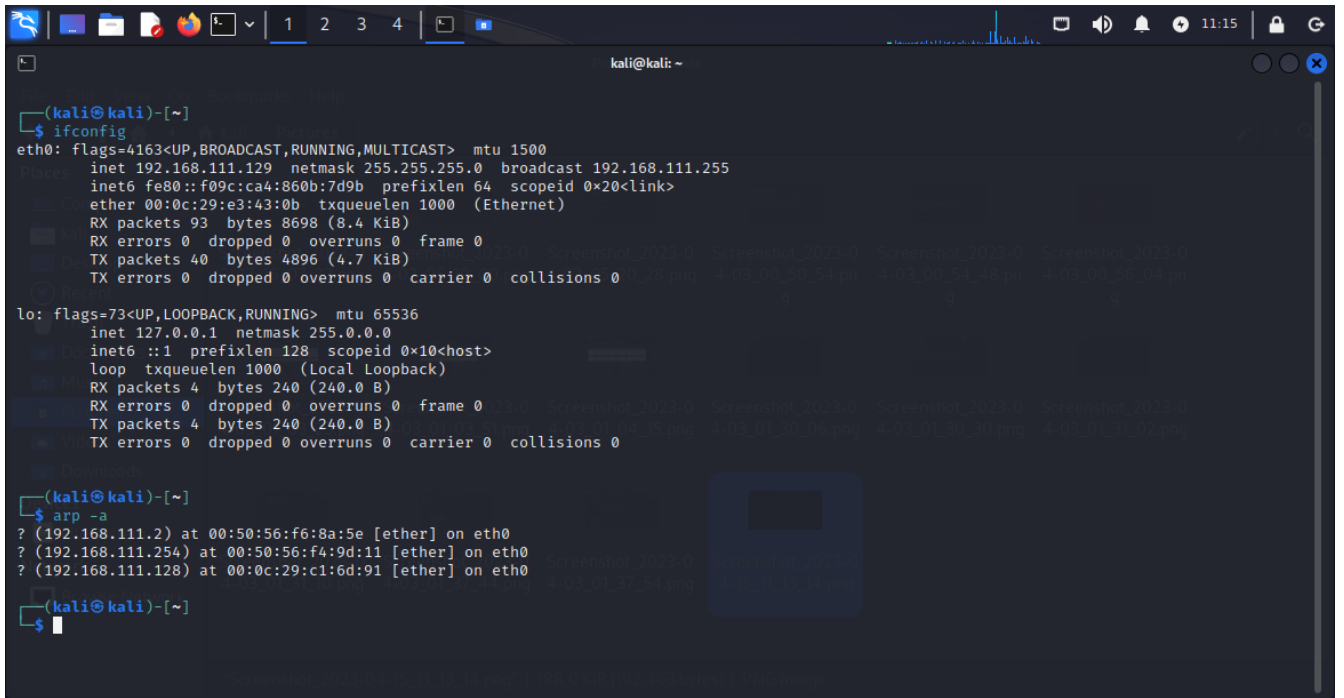
A screenshot of a Kali Linux terminal window. The terminal shows the output of the 'ifconfig' command for the 'eth0' and 'lo' interfaces. The 'eth0' interface is configured with IP 192.168.111.129, netmask 255.255.255.0, and broadcast 192.168.111.255. The 'lo' interface is configured with IP 127.0.0.1 and netmask 255.0.0.0. Below the 'ifconfig' output, the 'arp -a' command is executed, showing the ARP table with three entries: (192.168.111.2) at 00:50:56:f6:8a:5e [ether] on eth0, (192.168.111.254) at 00:50:56:f4:9d:11 [ether] on eth0, and (192.168.111.128) at 00:0c:29:c1:6d:91 [ether] on eth0. The terminal window has a title bar with 'kali@kali: ~' and standard window controls. The top of the screen shows a taskbar with various application icons and a system tray with a clock showing 11:15.

Figure 4.3: Victim Machine

4.2.2 Get User Response

When the user has given the input for checking the ARP table it will show the MAC addresses in the ARP table and IP addresses. The user will be using the Command Line Interface. The script could be run using a terminal or command prompt.

4.3 User Interface

The user interface's design rules for the project are as follows:

- * In order to improve usability, an interface needs to be well designed to be "user-friendly".
- * Use simple interfaces
- * Have extensive documentation

- * Good color scheme
- * User friendly
- * Easy to use
- * Simple and understandable

Component	Value
Typography	Monospace
Colors	Default: White text on black background Notifications: Red text on black background
Icon	Regular

Figure 4.4: User interface

User Interfaces The user will be using the Command Line Interface. The script could be run using a terminal or command prompt.

Hardware Interfaces The user can use any router and its dashboard for this project. There is no limitations as such.

Software Interface The Operating system used for this project is Linux and the programming language is Python. The tools that will be used are Wireshark, TShark and Tcpdump.

Communications Interfaces To send an alert to the alert, the user would be using the email as method of communication. Hence, the host “smtp.gmail.com” and port 465 (standard TCP) would be in use.

Chapter 5

Result and Discussions

The act of delivering malicious Address Resolution Protocol (ARP) packets to a LAN's default gateway in order to alter the Internet Protocol (IP) to Media Access Control (MAC) address table pairings is known as Address Resolution Protocol (ARP) spoofing. This kind of Cyberattack targets local area networks (LANs). Here, the administrator will be warned of the ARP spoofing attack through email and terminal. This will enable the administrator to lessen the amount of assaults and implement defences to stop them from occurring again in the future. Most businesses, no matter how big or little, have their own MAC filtering system where they would gather the system's MAC address and store it in a database. ARP spoofing may be done internally inside the organisation due to the fact that it is recorded in the database and is possible in the event of an intrusion. Therefore, this script will assist in identifying and notifying the Attack to stop it from happening.

Every communication that takes place will be recorded by the network traffic interceptor, which will also keep logs of it. a separate ARP traffic filter will separate out only the ARP traffics, making it simple to monitor.

Chapter 6

Deployment Process

6.1 Overview of the process

The UI used for this project is the command-line interface. A command-line interface (CLI) is a text-based user interface (UI) used to run programs, manage computer files and interact with the computer. CLIs accept as input commands that are entered by keyboard; the computer then runs the commands invoked at the command prompt. The project is designed in such a way that the UI and the other elements will be user friendly. The programming languages used for this project is Python. A command line shell, Bash is used to run the scripts. The benefits of using shell is that it enables users to write programs that can be run on the system from the command line, direct the output of one program to be the input for another program, and system variables which can be set at the command line, or the values of those variables displayed. The IP address, MAC address, and interface of the machines were first stored in the ARP cache table when communication between them began. Once the script has begun to run, it will launch two separate terminals, one of which will capture network traffic and

the other of which will only filter out ARP traffic alone.

The network traffic interceptor will be capturing from the interface, it will capture all the transactions or the network communications that is taking place and it would save it in a .pcap file. This .pcap file can be later be analyzed by an IT administrator or a soc analyst using various tools like tcpdump or tshark or wireshark.

For the purpose of capturing the traffic, those interceptors will be running in the background. The script is now being executed in real time in the background. This script will catch any attempted attacks on the machine from outsiders. Arpspoof is a program that we use to attack. The command to launch the attack is **arpspoof -i eth0 -t 192.168.111.129 -r 192.168.111.2**.

As soon as the attack was launched, a warning message stating that ARP spoofing had been discovered and the ARP cache should be checked was sent to the administrator's email address as well as a message on the victim's workstation informing them that they were the target of the attack and providing their MAC address. After the attack takes place we can now check the log files which is stored as .pcap file which can be analysed by using wireshark. In network traffic logs , there has a .pcap file along with the timestamp generated. using wireshark we can analyze the .pcap file. This will contain all the traffic captured. In the ARP traffic interceptor logs , it will filter out the ARP traffic alone

6.2 Algorithm Implementation

The current algorithm deals with 3 systems Victim, Attacker, and Gateway over the network connected through the wired connection. Attacker uses arpspoof tools to play the role of man in the middle like an intruder. Algorithm checks the ARP cache. With the help of Arpspoof tool, ARP poisoning is done between Victim and Gateway using arpspoof command. Now all the traffic is routed to an Attacker. It captures all kind of traffic including ARP request and ARP response packets flowing among hosts. The algorithm detects the spoof and alerts the administrator. Additionally, the network and ARP traffic is monitored and logged to a log file which will be saved in the server

6.3 Component testing

Component testing is a sort of software testing where each component is tested independently without integrating with other components. When regarded from the standpoint of architecture, it is also known as "module testing". Unit testing, program testing, and module testing are other names for component testing. Any software generally consists of a number of parts. These components are tested separately using component level testing. It's one of the black box testing methods that the QA Team conducts the most frequently.

6.3.1 Network Traffic Interception

Network traffic interceptor will Collect network traffic which is transmitting through the connection or the communication that is taking place and save the network traffic

logs to disk with the timestamp. the script needs administrative privileges it will start intercepting the network traffic.

Input:The user runs the main.sh script. The script fires up network traffic.sh which intercepts the network traffic and saves or logs it to a file

Output: The traffic from the network will be intercepted and will be logged to the disk for further analysis. Using wireshark, a network administrator can analyse this logs.

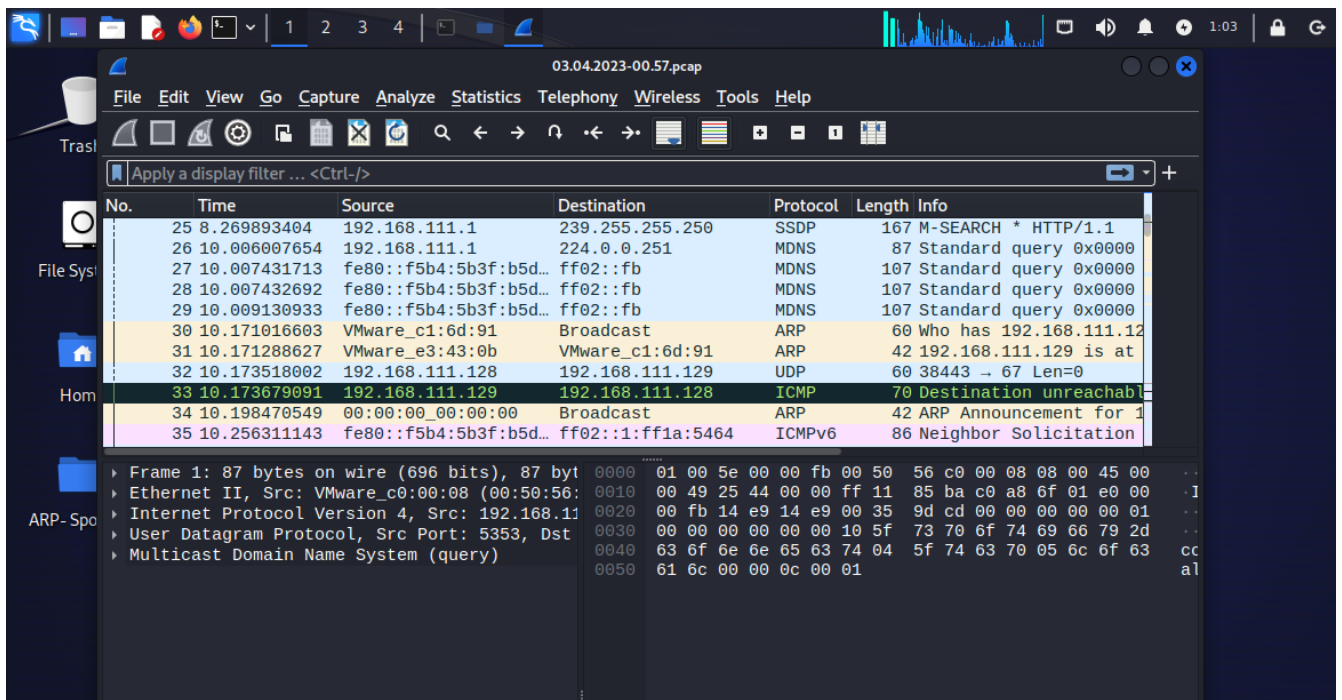


Figure 6.1: Network Traffic logs

6.3.2 ARP traffic Filter

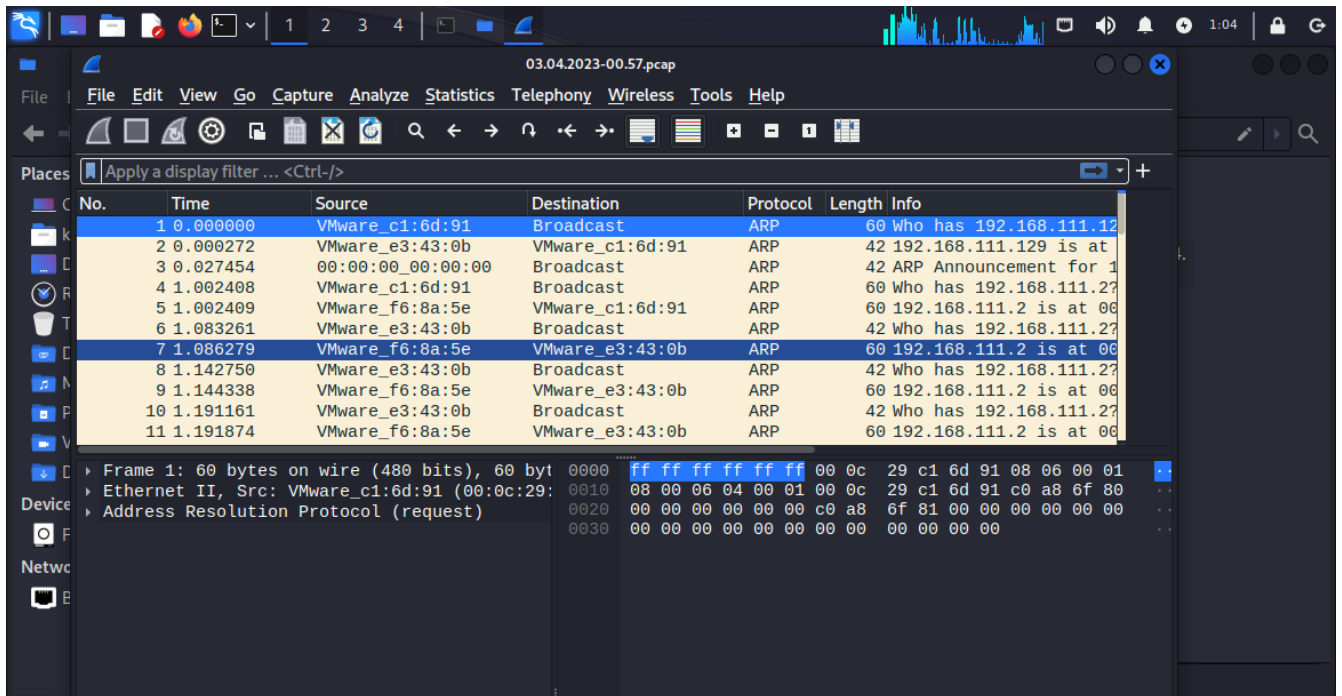
In a local network, the ARP protocol is used to translate IP addresses to MAC addresses.

You can see ARP requests and responses by capturing ARP traffic, which can assist you

in better understanding how devices communicate with one another in a network. The ARP traffic filter separates out the ARP traffic and saves the ARP traffic records with a timestamp. To filter the ARP traffic, the script requires sudo privileges. This ARP traffic will be used by the administrator which will be analysed by using wireshark.

Input: The user runs the main.sh script. The script fires up arp traffic.sh which filters the ARP traffic and saves or logs it to a file

Output: The traffic from the ARP will be filtered and will be logged to the disk for further analysis. Using wireshark, a network administrator can analyse this logs.



The image shows a Wireshark interface with a capture file named '03.04.2023-00.57.pcap'. The main pane displays a list of 11 ARP traffic packets. The columns shown are No., Time, Source, Destination, Protocol, Length, and Info. The packets are filtered to show only ARP traffic. The bottom pane shows the details of the selected packet (No. 1), including the Ethernet II header and the ARP request payload.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	VMware_c1:6d:91	Broadcast	ARP	60	Who has 192.168.111.12
2	0.000272	VMware_e3:43:0b	VMware_c1:6d:91	ARP	42	192.168.111.129 is at
3	0.027454	00:00:00_00:00:00	Broadcast	ARP	42	ARP Announcement for 1
4	1.002408	VMware_c1:6d:91	Broadcast	ARP	60	Who has 192.168.111.2?
5	1.002409	VMware_f6:8a:5e	VMware_c1:6d:91	ARP	60	192.168.111.2 is at 00
6	1.083261	VMware_e3:43:0b	Broadcast	ARP	42	Who has 192.168.111.2?
7	1.086279	VMware_f6:8a:5e	VMware_e3:43:0b	ARP	60	192.168.111.2 is at 00
8	1.142750	VMware_e3:43:0b	Broadcast	ARP	42	Who has 192.168.111.2?
9	1.144338	VMware_f6:8a:5e	VMware_e3:43:0b	ARP	60	192.168.111.2 is at 00
10	1.191161	VMware_e3:43:0b	Broadcast	ARP	42	Who has 192.168.111.2?
11	1.191874	VMware_f6:8a:5e	VMware_e3:43:0b	ARP	60	192.168.111.2 is at 00

Details of Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 Ethernet II, Src: VMware_c1:6d:91 (00:0c:29:00:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (request)

Figure 6.2: Filtered ARP Traffic logs

6.3.3 ARP Spoof Detector:

The ARP spoof detector script must have administrative privileges in order to begin monitoring ARP spoofing attempts. It will detect ARP spoof attacks that are transmitted across the communication channel. ARP spoof detector will scan the local area network (LAN) for any ARP packets and examine them to find any forged or suspicious packets.

Input: The user executes the script main.sh. The program launches arp detector.py, which scans for and recognises ARP spoofing threats.

Output: The script runs and sends an alert warning to the administrator if any ARP spoofing occurs. The administrator can then inspect the ARP cache, which contains the ARP traffic logs.

6.3.4 Alert Administrator

If an ARP spoofing attack is discovered, the script will send an email. If an ARP Spoof attack is discovered while the script is running with administrative rights, it will inform the administrator via email. This will make it easier to recognise when an ARP spoof occurs, and upon getting the email, the administrator may verify the collected ARP logs and take the necessary precautions to avoid them.

Input: The script main.sh is executed by the user. The program launches email alerts.py, which sends the administrator an email.

Output: If an ARP spoofing attack is discovered, an email alerting the administrator of the assault will be sent.

```
root@kali: /home/kali/Desktop/ARP- Spoofing
total 76
-rwxrwxrwx 1 kali kali 1394 Apr 2 13:17 arp_detector.py
drwxr-xr-x 2 root root 4096 Apr 2 13:38 arp_traffic_logs
-rwxrwxrwx 1 kali kali 260 Apr 2 13:17 arp_traffic.sh
-rwxrwxrwx 1 kali kali 10077 Apr 2 13:17 email_alerts.py
-rw-rw-rw- 1 kali kali 35149 Apr 2 13:17 LICENSE
-rwxrwxrwx 1 kali kali 228 Apr 2 13:17 main.sh
drwxr-xr-x 2 root root 4096 Apr 2 13:38 network_traffic_logs
-rwxrwxrwx 1 kali kali 182 Apr 2 13:17 network_traffic.sh
-rw-rw-rw- 1 kali kali 1879 Apr 2 13:17 README.md

(root@kali)-[/home/kali/Desktop/ARP- Spoofing]
# bash main.sh
# Option "--command" is deprecated and might be removed in a later version of gnome-terminal.
# Use "--" to terminate the options and put the command line to execute after it.
# Option "--command" is deprecated and might be removed in a later version of gnome-terminal.
# Use "--" to terminate the options and put the command line to execute after it.
/usr/lib/python3/dist-packages/scapy/layers/ipsec.py:462: CryptographyDeprecationWarning: Blowfish has been deprecated
cipher=algorithms.Blowfish,
/usr/lib/python3/dist-packages/scapy/layers/ipsec.py:476: CryptographyDeprecationWarning: CAST5 has been deprecated
cipher=algorithms.CAST5,

    "The quieter you become, the more you are able to hear"

Here's a list of interfaces in your system:

lo

eth0

Choose the desired interface: eth0

12:50:18_AM → You are under attack @ 00:0c:29:c1:6d:91!!
Traceback (most recent call last):
  File "/home/kali/Desktop/ARP- Spoofing/email_alerts.py", line 441, in <module>
```

Figure 6.3: Files of ARP Spoof detector

```
root@kali: /home/kali/Desktop/ARP- Spoofing

(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali]
# cd ARP-\ Spoofing
cd: no such file or directory: ARP- Spoofing

(root@kali)-[/home/kali]
# cd Desktop/ARP-\ Spoofing

(root@kali)-[/home/kali/Desktop/ARP- Spoofing]
# ./main.sh
# Option "--command" is deprecated and might be removed in a later version of gnome-terminal.
# Use "--" to terminate the options and put the command line to execute after it.
# Option "--command" is deprecated and might be removed in a later version of gnome-terminal.
# Use "--" to terminate the options and put the command line to execute after it.
/usr/lib/python3/dist-packages/scapy/layers/ipsec.py:462: CryptographyDeprecationWarning: Blowfish has been deprecated
cipher=algorithms.Blowfish,
/usr/lib/python3/dist-packages/scapy/layers/ipsec.py:476: CryptographyDeprecationWarning: CAST5 has been deprecated
cipher=algorithms.CAST5,

Here's a list of interfaces in your system:

lo

eth0

Choose the desired interface: 
```

Figure 6.4: ARP Spoof detector in Victims machine

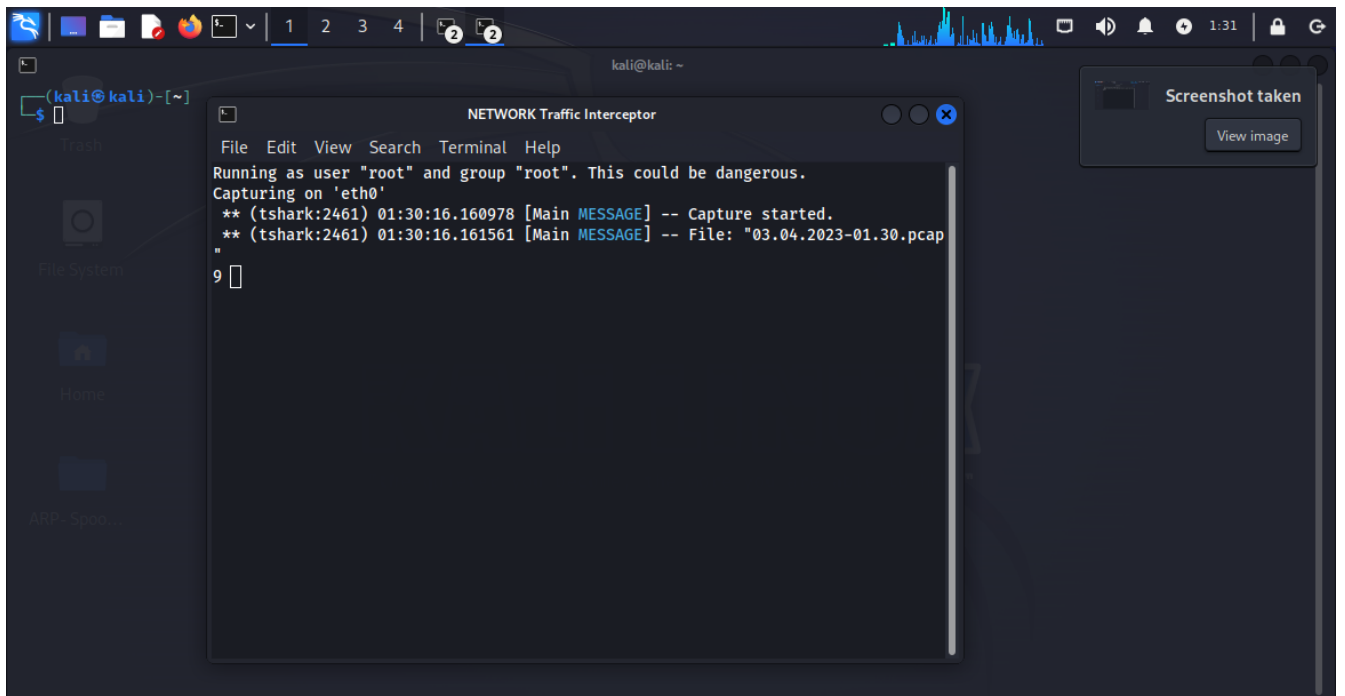


Figure 6.5: network Traffic Interceptor

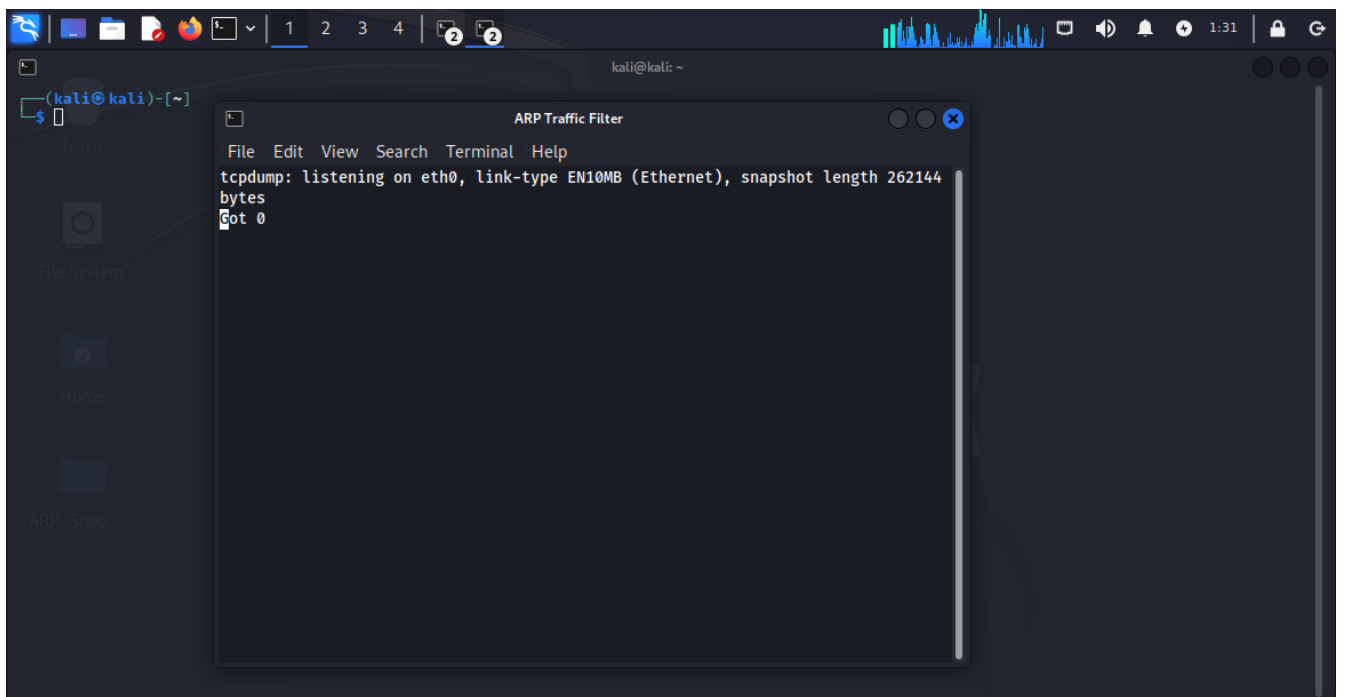


Figure 6.6: ARP Traffic Interceptor

```
(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali]
# arpspoof -i eth0 -t 192.168.111.129 -r 192.168.111.2
0:c:29:c1:6d:91 0:c:29:e3:43:b 0806 42: arp reply 192.168.111.2 is-at 0:c:29:c1:6d:91
0:c:29:c1:6d:91 0:50:56:f6:8a:5e 0806 42: arp reply 192.168.111.129 is-at 0:c:29:c1:6d:91
^Z
zsh: suspended arpspoof -i eth0 -t 192.168.111.129 -r 192.168.111.2
(root@kali)-[/home/kali]
#
```

Figure 6.7: Attackers machine

```
Here's a list of interfaces in your system:

    lo
    eth0

Choose the desired interface: eth0
12:54:38_AM -> You are under attack @ 00:0c:29:c1:6d:91!!
```

Figure 6.8: Alert on terminal

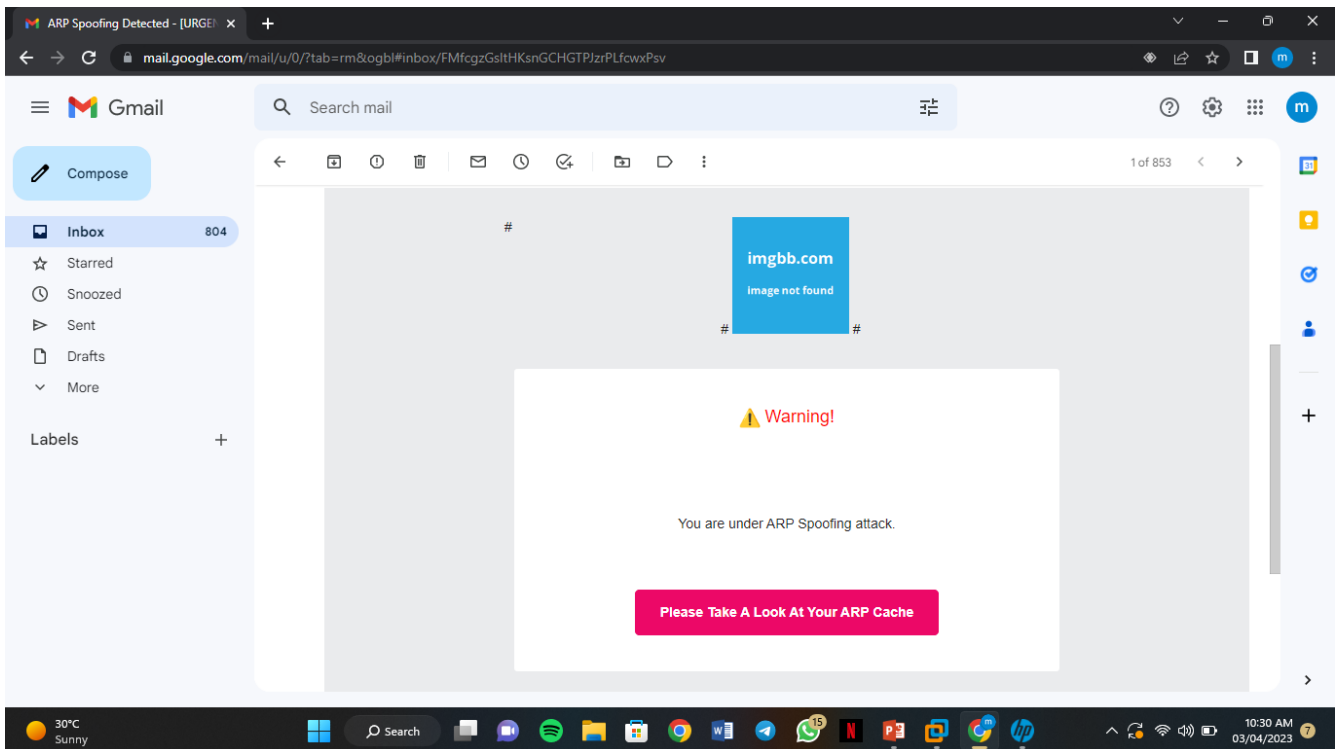


Figure 6.9: Alert via Email

Chapter 7

Conclusion

In this effort, an active method to identify ARP spoofing was developed. It was demonstrated that the method employed here is more quicker and more adaptable than passive detection methods. The time lag between learning new addresses and spotting spoofing is as little as possible since active approach is employed here to investigate the authenticity of ARP traffic on a per packet basis. Despite the numerous fixes offered, brand-new attack methods could nonetheless result in brand-new security issues because the ARP protocol has a few fundamental flaws. Therefore, more research is needed to address the ARP protocol's underlying security flaws. This will show us the ARP traffic and how the attack is happening and by the scripts which the alert notification send to the administrator, this can prevent the ARP spoofing attack from happening. The user will receive an alert by telling the system is under ARP spoofing attack and they terminate the particular session by disconnecting the connection

7.1 Future Works

The addition of newer, more modern, and more sophisticated features to this project will be made possible by future research, studies, and advancements in the field of ARP spoofing. The tool can be used in a variety of circumstances. Although challenging, designing a GUI is not impossible.

References

1. Sherin Hijazi, Mohammad S. Obaidat. A New Detection and Prevention System for ARP Attacks Using Static Entry , IEEE Systems Journal Volume: 13, Issue: 3 , Sept 2019
2. Mohammad Z. Masoud, Yousf Jaradat and Ismael Jannoud , On Preventing ARP poisoning attack utilizing software defined network (SDN) paradigm , Nov 2015
3. <https://linuxhint.com/wireshark-command-line-interface-tshark/>
4. <https://www.hindawi.com/journals/scn/2022/2196998/>
5. Mahesh V. Tripunitara, Partha Dutta. A Middleware Approach to Asynchronous and Backward Compatible Detection and Prevention of ARP Cache Poisoning
6. Sumit Kumar, Shashikala Tapaswi. A Centralized Detection and Prevention Technique against ARP Poisoning , : July 2012
7. <https://www.researchgate.net/publication/329078115> A New Detection and Prevention System for ARP Attacks Using Static Entry

8. Haider Salim; Zhitang Li, Hao Tu, Zhengbiao Guo. Preventing ARP Spoofing Attacks through Gratuitous Decision Packet, IEEE, Oct 2012
9. Seung Yeob Nam, Dongwon Kim, Jeongeun KimEnhanced. ARP: Preventing ARP Poisoning-Based Man-in-the-Middle Attacks , IEEE Communications Letters (Volume: 14 , Issue: 2 , February 2010)
10. Daesung Moon , Jae Dong Lee, Young-Sik Jeong ,Jong Hyuk Park . RTNSS: a routing trace-based network security system for preventing ARP spoofing attacks , The Journal of Supercomputing volume 72, pages1740–1756(2016) , Dec 2014
11. Nikhil Tripathi, BM Mehtre, Analysis of Various ARP Poisoning Mitigation Techniques: A comparison July 2014
12. Seung Yeob Nam ,Sirojiddin Djuraev , Minho Park . Collaborative approach to mitigating ARP poisoning-based Man-in-the-Middle attacks , Elsevier – Computer networks Volume 57 Dec 2013
13. Seung Yeob Nam , Sirojiddin Jurayev, Seung-Sik Kim, Kwonhue Choi and Gyu Sang Choi ,Mitigating ARP poisoning-based man-in-themiddle attacks in wired or wireless LAN , EURASIP Journal on Wireless Communications and Networking volume 2012
14. A Holistic Approach to ARP Poisoning and Countermeasures by Using PracticalExamples and Paradigm, Faisal Rahman, P. Kamal Computer Science 2014

15. A New Detection and Prevention System for ARP Attacks Using Static Entry
Sherin Hijazi, M. Obaidat Computer Science IEEE Systems Journal 2019
16. X. Hou Z. Jiang and X. Tian "The Detection and Prevention for ARP Spoofing based on SNORT" Proc. of IEEE International Conference on Computer Application and System Modeling (ICCASM'10) vol. 5 pp. 5-137.